(BioMagResBank, 2004). The *starlib* class library was developed at BMRB for handling NMRStar files, but its initial application to such files independently of the prototype data dictionary means that it is applicable to any STAR File. It does not provide a relational database paradigm (although this is a long-term goal). However, it does provide objects and methods suitable for searching and manipulating STAR data.

Table 5.2.6.1 lists the top-level classes used in *starlib*. *ASTnode* is a formal base class, providing the types and methods that can be used in other derived classes. *StarFileNode* is the root parent of all other objects contained in an in-memory representation of a STAR File; in practice it contains a single *StarListNode*, which is the list of all items contained in the file. *BlockNode* is a class which contains a partition of the STAR File: the class handles both data blocks and global blocks. Data-block names are stored in instances of the *HeadingNode* object, which also holds save-frame identification codes and is therefore useful for accessing named portions of the file.

DataNode is a virtual class representing the types of data objects handled by the library (accessed directly as *DataItemNode*, *DataLoopNode* and *SaveFrameNode*).

Looped data items are handled by a number of objects. *DataLoopNameListNode* is a list of lists of names in a loop. The first list of names is the list of names for the outermost loop, the second list of names is the list of names for the next nesting level and so on. *LoopNameListNode* is a list of tag names representing one single nesting level of a loop's definition. *LoopTableNode* is a table of rows in a *DataLoopNode* (not itemized in Table 5.2.6.1; it is an object representing a list of tag names and their associated values, a particular case of *DataNode*). *starlib* views a loop in a STAR file as a table of values, with each iteration of the loop being a row of the table. Each row of the table can have another table under it (another nesting level), but such tables are the same structure as the outermost one. Thus *LoopTableNode* stores a table at some arbitrary nesting level in the loop. A simple singly nested loop will have only one loop table node, but a multiply nested loop will have a whole tree of loop tables. *LoopRowNode* is a single row of values in a loop.

DataNameNode holds the name of a tag/value pair or a loop tag name. *DataValueNode* is the type that holds a single string value from the STAR file and the delimiter type that is used to quote it.

DataListNode and *SaveFrameListNode* store lists of data within higher-order data objects or save frames, and are internal classes rarely invoked directly by a programmer.

A number of observations may be made regarding this approach. Firstly, the objects can be mapped with reasonable fidelity to the high-level Backus–Naur form representation of STAR (Chapter 2.1). Secondly, it is computationally convenient to abstract common features into parent classes, so that, for example, individual data items, looped data and save frames are represented as child objects of the *DataNode* object, and not themselves as first-generation children of the base class. Thirdly, the handling of nested loops may be achieved in different ways; *starlib* has chosen a particular view that is perhaps well suited to relational data models.

As expected within a programming toolkit, *starlib* offers a large number of methods for retrieving STAR data values, adding new data items, extending or re-ordering list structures, and performing structural transformations of the in-memory data representation. Unlike the stand-alone *Star_Base* application, it does not guarantee that output data will be in a STAR-conformant format; and the programmer is left with the responsibility of validating transformed data at a low level.

Table 5.2.6.1. *Object classes for manipulating STAR data in starlib*

ASTnode	The base class from which all other classes are derived
StarFileNode	The STAR File object
StarListNode	List of items contained in the STAR File
BlockNode	A data or global block
HeadingNode	Labels for major STAR File components
DataNode	General class for data objects
DataLoopNameListNode	List of lists of names in a loop
LoopNameListNode	List of tag names representing one nesting loop level
LoopTableNode	Table of rows in a loop
LoopRowNode	Single row of values in a loop
DataNameNode	A data name
DataValueNode	A single string value
DataListNode	List of data within a higher-order data object
SaveFrameListNode	List of data items allowed in a save frame

Nevertheless, this is a substantial and important library which, as with *CIFOBJ*, has played an important role in the functioning of a major public data repository. Development of the class libraries continues, with a Java version now available.

5.2.6.5. *StarDOM*

A convenience of well designed object representations is that effective transformation between different data representations may be possible. The *StarDOM* package (Linge *et al.*, 1999) demonstrates a transformation from STAR Files to an XML representation, where the tree structure of a STAR File as interpreted in the *starlib* view above is mapped to a document object model (DOM; W3C, 2004). This approach is similar to *Jumbo*, mentioned above in Section 5.2.6.2.

A demonstration of *StarDOM* is the transformation of the complete set of NMR data files at BioMagResBank to XML. The resultant files can then be interrogated using the *XQL* query language (Robie *et al.*, 1998). In this example implementation, the target XML document type definition (DTD) includes a small number of XML elements matching the STAR objects *global* and *data block*, *save frame*, *list*, *data item*, *data name* and *data value*. Particular data names are recorded as values of the <NAME> element. The authors of the *StarDOM* package are considering an extension in which named data items map directly to separate XML elements; the goal is to develop an NMR-specific DTD that is isomorphous to the emerging NMRStar data dictionary.

References

Berman, H. M., Battistuz, T., Bhat, T. N., Bluhm, W. F., Bourne, P. E., Burkhardt, K., Feng, Z., Gilliland, G. L., Iype, L., Jain, S., Fagan, P., Marvin, J., Padilla, D., Ravichandran, V., Schneider, B., Thanki, N., Weissig, H., Westbrook, J. D. & Zardecki, C. (2002). *The Protein Data Bank. Acta Cryst.* D**58**, 899–907.

Berman, H. M., Westbrook, J., Feng, Z., Iype, L., Schneider, B. & Zardecki, C. (2002). *The Nucleic Acid Database. Acta Cryst.* D**58**, 889–898.

BioMagResBank (2004). *NMR-STAR Dictionary*. Version 3.0. (Under development.) http://www.bmrb.wisc.edu/dictionary/3.0html/.

Chang, W. & Bourne, P. E. (1998). *CIF applications. IX. A new approach for representing and manipulating STAR files. J. Appl. Cryst.* **31**, 505–509.

Knuth, D. E. (1973). *The art of computer programming.* Vol. 3, *Sorting and searching*, pp. 422–447. Reading, MA: Addison-Wesley.

Linge, J. P., Nilges, M. & Ehrlich, L. (1999). *StarDOM: from STAR format to XML. J. Biomol. NMR*, **15**, 169–172.

Murray-Rust, P. (1993). *Analysis of the DDL/dictionary parsing problem. Proc. First Macromolecular Crystallographic Information File (CIF) Tools Workshop*, ch. 12. New York: Columbia University.